EXHIBIT 6

REDACTED

IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF VIRGINIA
ALEXANDRIA DIVISION

UNITED STATES OF AMERICA, et al., Plaintiffs, v. GOOGLE, LLC, Defendant.	Case No. 1:23-cv-00108 (LMB/JFA) HON. LEONIE H. BRINKEMA

EXPERT REPLY REPORT OF GORANKA BJEDOV

August 11, 2025

strategies that enable safe transitions without service interruption, such as canary rollouts and shadow testing.[7]

8. I further demonstrate that my plan accounts for the technical complexity of AdX and DFP throughout each stage of the migration process in Section III.B. This includes the work involved in mapping and replacing proprietary dependencies, isolating services, modifying integration points, and validating performance at each stage.

9. I also describe why the approach I propose in my migration plan is effective for systems that contain interdependencies between microservices in Section III.B.2. Despite Mr. Maymudes' claims, interdependent systems, which my plan and timeline account for, can be developed and migrated in parallel using the coordinated migration approach and timeline I propose using techniques such as dependency mapping, creating mocks and stubs, and staging gradual deployments of replacement services.[8]

10. In addition, I clarify the preconditions assumed in my proposed plan and timeline in Section III.C. These preconditions are consistent with industry practice and reflect the natural boundary between legal and business activities and technical execution. The process of actually migrating AdX and DFP would not begin until an acquirer is identified.[9] However, some of the activities of my proposed migration plan may begin while legal and business activities are ongoing and before an acquirer is identified, such as demarcating an initial scope of code to be migrated, creating a wrapper to isolate the system calls to proprietary services that would be replaced to set up for potential alternative system options, and performing parts of the deployment analysis.

11. In response to the industry case study examples I discuss in Section V of my Opening Report, Dr. Nieh opines that migrations and deployments are different things, and as a result deployment work

[7] *See* Expert Report of Goranka Bjedov, PhD, United States of America, et al., v. Google, LLC, Case No. 1:23-cv-00108, July 7, 2025 ("Bjedov Opening Report"), ¶ 189 ("Each service can be analyzed in parallel by its own two-to-three-person engineering pod. Pods may own the code inspection, build a stub (or full replacement), define tests (see Section IX.D), and shepherd the change through canary and production ramps.") and ¶ 194 ("The actual mechanics for this process are straightforward. Every new service is launched with three settings. First, it is *disabled*, which means the system is still using a test version. Second, it moves to *shadow mode*, where real traffic is sent to the new service behind the scenes, but its answers aren't actually used") (emphasis in original).

[8] Bjedov Opening Report, Section IX.C.

[9] However, additional migration work could begin without the identification of an acquirer if, for whatever reason, the target environment for the migration (e.g., GCP) has already been determined. In this case, Google could choose replacements for its services with the best alternatives from the landing zone.

is not comparable to work that is done in a migration.[10] In doing so, Dr. Nieh ignores the fact that deployments and migrations have a close relationship and share many of the same considerations. Dr. Nieh supports his point through the use of a nonstandard definition of "software migration," which requires moving software to an entirely new environment, implying that internal software migrations are not, as the name suggests, true software migrations.[11] This is not how the term is used in large-scale engineering organizations, including Google,[12] and there are many shared activities between internal migrations and external migrations, as I describe in Section IV.A.

12. Mr. Maymudes claims my analysis of AdX and DFP's migration is incomplete by ignoring the fact that the very work he claims to be missing from my analysis is present in Dr. Weissman's report,[13] which served as the foundation for my migration assessment in my Opening Report.[14] Because of this, he mischaracterizes the scope of my report and professional experience as a Site Reliability Engineer, which is directly relevant to the migration of AdX or DFP. I discuss Mr. Maymudes' criticism in Section IV.B.

13. Mr. Maymudes implies that my background in deployment and scaling from my career as a performance and capacity engineer is insufficient or irrelevant to evaluating software migration

[10] Nieh Rebuttal Report, ¶ 91 ("Dr. Bjedov refers to certain projects or work as large-scale software migrations in order to suggest they are comparable to the type of migration required for an AdX or DFP divestiture, but what she is actually referring to are software *deployments*. That distinction is critical because deployments do not involve the same considerations and technical complexity that a migration of AdX or DFP would") (emphasis in original).
[11] Nieh Rebuttal Report, ¶ 92 ("Deploying software refers to releasing software, which in the context of Google's ad tech would generally mean releasing a new feature that runs on Google's internal infrastructure, i.e., there is no migration to a completely new or different infrastructure that lacks the full set of proprietary dependencies available as part of Google's internal infrastructure. In contrast, a large-scale software *migration* in the context of a divestiture refers to moving software to a new environment, and all the work required to make the migrated software operational in that new environment") (emphasis in original).
[12] Maymudes Rebuttal Report, ¶ 8 ("From 2013 to 2018, I was Technical Lead for another large-scale migration known internally as 'Skyray,' which involved migrating Xbid onto infrastructure shared with Google Ads, internally referred to as the CAT2 serving stack. The Skyray migration was entirely within Google's production environment, such that Xbid could rely on Google's infrastructure systems both before and after the migration.").
[13] Maymudes Rebuttal Report, ¶ 29 ("Engaging with the complexity of a divestiture of DFP or AdX requires an analysis that is grounded in understanding the complexity and scale of Google's ad tech systems from at least both the perspectives of Software Engineering and Site Reliability Engineering. However, Dr. Bjedov appears to narrowly frame her opinions based on her experience in SRE-adjacent roles. She primarily focuses on certain aspects of scale in her evaluation of a potential divestiture of DFP or AdX, and does not holistically address the complexity of developing and maintaining large-scale, distributed software systems throughout their software lifecycles.").
[14] Weissman Opening Report, ¶ 2 ("I have been asked to assess whether it is feasible, from a technical perspective, to copy and migrate certain technical assets, such as repositories of source code, configuration files, and data, from Google's Ad Exchange ('AdX') and current publisher-ad server ('DFP'), including the final auction logic and remainder of DFP. I have been asked to assess the technical feasibility of developing or adapting application programming interfaces ("APIs") that facilitate the proposed data sharing and interim remedies.").

feasibility.[15] This reflects a misunderstanding of the role of SREs, who are responsible for ensuring the reliable operation of complex, production-scale systems, especially during periods of change such as migrations. My responsibilities included managing migrations of systems at scale, including performance risk, incident mitigation, service decompositions, and gradual rollouts. These skills and responsibilities are directly applicable to the migration of AdX and DFP. I further discuss Mr. Maymudes' misunderstandings in Section V.

14. Dr. Nieh opines that some of the tools I list in my Opening Report are not relevant to the software migrations of AdX or DFP.[16] Dr. Nieh misunderstands the purpose of my examples, which is to show that there are tools available to assess the migration complexity of software applications and not necessarily to prescribe the tools an acquirer must use, as it is ultimately up to the acquirer to select the ideal tools for their use. I discuss Dr. Nieh's argument and the relevance of several examples of tools in Section IV.C below.

15. Dr. Nieh and Mr. Maymudes suggest that AdX and DFP would need to be substantially rewritten to function outside of Google's internal infrastructure.[17] I disagree. While some application code and configuration updates are expected when replacing internal dependencies (i.e., proprietary Google systems), these do not amount to a wholesale rewrite. This type of modification, which includes updating API calls and swapping service endpoints, is routine in migrations and is accounted for in my timeline. I discuss this further in Section III.B.

16. In their Rebuttal Reports, Dr. Nieh and Mr. Maymudes claim that the industry examples I discuss in my Opening Report are either not relevant to this migration or support their view that the migration of AdX and DFP will take many more years than I propose.[18] In particular, they dismiss my case studies due to differences in scale. I explain in Section V.B that scale is not the only factor

[15] Maymudes Rebuttal Report, Section II.A.
[16] Nieh Rebuttal Report, ¶ 132 ("In her report, Dr. Bjedov purports to identify tools that would facilitate a speedier divestiture. But many of the tools she refers to are irrelevant to software migrations of the kind required here; she misunderstands how those tools work and why they are useful.").
[17] Expert Report of Jason Nieh, PhD, United States of America, et al., v. Google, LLC, Case No. 1:23-cv-00108, July 7, 2025 ("Nieh Opening Report"), ¶ 18 ("The products that Plaintiffs propose that Google divest do not currently exist, and completely new versions would have to be created solely to be sold."); Maymudes Rebuttal Report, ¶ 32 ("As a result, the Bjedov Report largely ignores the greatest source of complexity in a DFP or AdX divestiture: the fact that the existing source code, even if recompiled, simply will not function outside of Google's proprietary production computing environment and would thus need to be substantially rewritten before it could be 'migrated' to a 'typical cloud environment.'").
[18] *See* Maymudes Rebuttal Report, Section IV; *see also* Nieh Rebuttal Report, Section V.

that makes a case study relevant, and what matters is how the systems involved in each case study are architected and whether the systems demonstrate near-linear scalability, horizontal elasticity, and successful risk management. I further discuss that the migration examples in my Opening Report all involved complex distributed systems and show that migrations of comparable difficulty can be completed within reasonable timeframes when executed in accordance with best practices.

17. In their Rebuttal Reports, Dr. Nieh and Mr. Maymudes conflate unrelated projects and corporate activities with the software migrations at issue in this case. By including extraneous steps, such as additional product integrations, platform consolidation, and continued feature development after launch, they artificially extend the perceived timeline and complexity of these migrations beyond what is technically required for a lift-and-shift transition. This mischaracterization artificially increases the perceived length of the migrations and undermines a clear assessment of the actual engineering effort involved. I discuss this further in Section V.A.

18. For example, Mr. Maymudes highlights the Invite Media and Project Skyray migrations as analogues to the AdX and DFP migrations. However, both migrations involved full-stack integrations (i.e., efforts to merge acquired products into Google's broader ecosystem, including alignment with Google's infrastructure, data models, and user interfaces). Invite Media, in particular, required a complete redesign of its functionality into a unified Google stack while maintaining parallel development on its original platform. Similarly, Project Skyray sought to consolidate backend systems across multiple products, requiring changes not only to infrastructure but also to logic and customer-facing features. These types of integration efforts are materially different from the technical migrations I propose for AdX and DFP, which assumes a lift-and-shift approach to a cloud-based environment like Google Cloud Platform ("GCP"). The migration plan that I propose does not require reimplementation within an acquirer's existing tech stack or functional integration with other systems. Instead, it focuses on replacing internal dependencies where necessary and validating performance in the new environment, which are significantly less complex than full-stack integration. I discuss this further in Section V.C.

19. Similarly, Dr. Nieh and Mr. Maymudes both point to the Gmail migration as the most relevant industry example due to the scale of the data involved.[19] I agree that the Gmail migration example represents a useful upper-bound benchmark, but their comparison fails to account for key differences in engineering context and business constraints. As I discuss in Section V.B, the complexity of Gmail migration arose in part from Gmail-specific requirements, such as validating objects account-by-account and maintaining parallel production environments over several years to comply with consumer privacy guarantees. These requirements are not present in the migrations of AdX and DFP. The acquirer could choose to stage or omit the migration of older non-critical datasets, focus only on functionality-critical data, and leverage modern tools for replication and schema conversion. Thus, while the Gmail example is informative, it is not determinative of the difficulty or timeline of the AdX or DFP migrations. In Section V.B, I further discuss other mischaracterizations that Dr. Nieh and Mr. Maymudes make that artificially inflate the perceived difficulty of the industry examples.

20. Dr. Nieh argues that migrating AdX or DFP to GCP would be infeasible due to differences between Google's internal environment and GCP. This overlooks key facts. GCP mirrors many aspects of Google's internal systems and is already used to host large-scale, latency-sensitive products.[20] [redacted].[21] The high degree of infrastructure overlap makes GCP a logical and efficient migration target.[22]

21. Nothing in Dr. Nieh's Rebuttal Report, Mr. Maymudes' Rebuttal Report, or in the documents cited therein changes the opinions I provide in my Opening Report or Rebuttal Report.

---

[19] *See* Maymudes Rebuttal Report, ¶ 163 ("Relative to the other case studies Dr. Bjedov cites, the Gmail migration provides a better estimate of how challenging it would be to divest AdX and DFP because it involves a product of more comparable scale and more similar infrastructure considerations."); *see also* Nieh Rebuttal Report, ¶ 125 ("Dr. Bjedov's Gmail migration case study is the only one of more similar scale to what would be required to divest AdX or DFP today, but even that case study involved a partial migration of less complexity—i.e. of one particular dependency, Gmail's database storage system, *within* Google's internal infrastructure (from Megastore to Spanner) and not of all dependencies *outside* Google's internal infrastructure") (emphasis in original).

[20] *See* Section III.B.2.b.i.

[21] *See generally* GOOG-AT-MDL-B-009828652; GOOG-AT-MDL-006926152; *see also* Bjedov Opening Report ¶ 162 (discussing GOOG-AT-MDL-B-009828652).

[22] *See* Bjedov Opening Report, Section VI.A ("The Similarity of Google Cloud Platform's Infrastructure to that of Google's Private Cloud Facilitates Software Migration Between the Systems").

that organizations using these patterns experience sharply lower incident rates and seamless user experience during migrations.[48] I also describe other risk-mitigation techniques including repeated rollback and monitoring checkpoints, which enable such zero-disruption migrations when testing new services.[49] Dr. Nieh labels these steps as "missing,"[50] "repeated,"[51] "illogical" and "disruptive,"[52] but these steps are all well-established software migration techniques that take place throughout the migration process. The structure I provide maps onto other migration blueprints used by major cloud providers and hyperscalers that regularly conduct migrations.[53]

B. My Plan Accounts for the Technical Demands of Migrating AdX and DFP

34. I agree with Dr. Nieh and Mr. Maymudes that planning and executing a software migration requires effort and skill.[54] Migrating AdX and DFP will not be a trivial undertaking, which is why I estimate

---

bugs into the code? A. Same as every -- every project. Starting from, you know, style guides, you know, best practices, unit tests, stress test"). Migrations are steady, progressive processes at any scale. *See* Deposition of Noam Wolf (Google), July 18, 2025, 83:18–84:13 ("A….No large-scale migration is ever an immediate cutover. In fact, no small -- not even small serving changes are one -- are, like, zero to a hundred.").

[48] *See, e.g.*, Dustin Smith, Daniella Villalba, Michelle Irvine, Dave Stanke, and Nathen Harvey, "State of DevOps 2021," Devops Research and Assessment, p. 9, https://dora.dev/research/2021/dora-report/2021-dora-accelerate-state-of-devops-report.pdf (comparing elite teams' change-failure rate at 0–15% to low performing teams' change-failure rate at 16–30%; elite teams have roughly 3× fewer incidents); *see also* "Blue-Green Deployment for Zero Downtime Releases," GTS, accessed August 8, 2025, https://globaltechnosol.com/case_studies/blue-green-deployment-for-zero-downtime-releases/ ("[Blue-green deployment strategy resulted in] zero downtime achieved…Deployment time was reduced by 40%…10% increase in user retention…reduced production errors by 30%…rollback times cut to under 5 minutes.").

[49] During migrations Google has performed, bugs still occur despite prevention efforts. While some bugs will always be present, this does not have to lead to a major outage necessarily. *See* Deposition of Noam Wolf (Google), July 18, 2025, 86:3–22 ("Q. Are those fairly effective measures in mitigating the risk of introducing bugs into the software code? A. Again, like, we write code, so we write bugs…We've had very, very little downtime over the course of the time that I've been managing this product, and I think that goes to kind of prevention. But bugs are a daily occurrence.") and 116:21–117:6 ("Q. Did the new publisher ad server code rewritten to Google system experience any major outages during or after the migration? A. I don't remember major things. I mean, I know that we had bugs, like any software would have…Just kind of procedural migration stuff.").

[50] Nieh Rebuttal Report, ¶ 166.

[51] Nieh Rebuttal Report, ¶ 140.

[52] Nieh Rebuttal Report, ¶ 139 ("As Tables IX.1 and IX.2 currently stand, they present a migration plan that is not viable, and not only would not result in a divested AdX or DFP, but also would cause serious disruption to customers during the migration process. Thus, because Dr. Bjedov's timeline estimates rely on her migration plan, the estimates are also unreliable for the additional reason that the underlying migration plan is illogical.").

[53] *See, e.g.*, "Migrate to Google Cloud: Best Practices for Validating a Migration Plan," Google Cloud, last modified May 5, 2025, https://cloud.google.com/architecture/migration-to-google-cloud-best-practices ("This document describes the best practices for validating the plan to migrate your workloads to Google Cloud.").

[54] Maymudes Rebuttal Report, ¶ 40 ("In my experience, developing a complex enterprise software project (or substantially changing the environment or requirements of an existing project) involves multiple stages, including: gathering business and regulatory requirements, evaluating necessary dependencies, creating high-level designs, writing/refactoring code, testing code, creating a deployment plan, deploying the code, scaling that deployment, and onboarding customers."); Nieh Opening Report, ¶ 159 ("Based on the requirements of the AdX and DFP

18 months and 24 months, respectively, to complete the migrations. My plan explicitly incorporates the complexity of these systems throughout each phase of the migrations and reflects an industry-standard, phased approach that is supported by modern tools to manage risks. In this section, I elaborate on how the migration processes and timelines I describe account for the specific demands of migrating AdX and DFP, which Mr. Maymudes and Dr. Nieh overlook.

35. With the resources described in the proposed timeline, such as an experienced team, full access to AdX and DFP system documentation and configuration files, and access to Google engineers,[55] I stand by my original assessment that the deployment analysis phase of each migration can be completed in one month.[56]

1) My Plan Accounts for the Technical Demands of Migrating AdX and DFP

36. Mr. Maymudes and Dr. Nieh claim that I do not consider the complexity of AdX and DFP in my timeline. They disregard[57] the factors I explicitly address in Section IV.B.3 ("There Are Many Factors That Influence the Complexity of the Migration and Deployments of Software Applications"[58]) of my Opening Report, as well as distinctions throughout my report.

a) I Consider Complexity in My Assessment of the Effort to Migrate Dependencies in AdX and DFP

functionalities (just a subset of which were outlined above) and the complex, sophisticated hardware and software infrastructure on which they operate, I expect producing an architecture development plan would take a considerable amount of time.").

[55] Bjedov Opening Report, ¶ 115 ("Google maintains internal documentation and tooling that maps specific services to team ownership, such that distinct engineering groups are responsible for deploying, monitoring,and maintaining particular services or jobs."), ¶ 73 ("Therefore, having appropriately organized teams of around 6-8 experienced engineers is the best combination for efficient project management."), and ¶ 177 ("This stage will require collaboration between software engineers at Google and the acquirer's company. Early technical collaboration with Google engineers and allowing the acquirer's engineers access to interface specifications is essential to draw clean system boundaries and avoid late-stage surprises.").

[56] Bjedov Opening Report, Figures IX.1 and IX.2, showing that I estimate "Stage 1: Deployment Analysis" phase will require one month.

[57] Mr. Maymudes criticizes that I do not explicitly list the similarity of the initial environment and the target environment. *See* Maymudes Rebuttal Report, n. 24 ("*See* Bjedov Report, § IV.B.3 (not listing these factors in the 'many factors that influence the complexity of the migration and deployments of software applications')."); I will reiterate my point that this was not intended to be an exhaustive list. *See* Bjedov Opening Report, ¶ 64 ("This is not an exhaustive list"); furthermore, I discuss the importance of the similarity between the initial and target environment. *See* Bjedov Opening Report, Section VI.A ("The Similarity of Google Cloud Platform's Infrastructure to that of Google's Private Cloud Facilitates Software Migration Between the Systems").

[58] *See generally* Bjedov Opening Report, Section III.

37. Mr. Maymudes and Dr. Nieh both criticize my proposed timeline for not considering the complexity of microservices that support DFP and AdX. Mr. Maymudes states that I "suggest that Google's software systems and services are designed as microservices, or small, independent, and loosely coupled services, that can be 'reroute[d] . . . across different environments without changing their functionality.' Based on my years of experience as a software engineer in the adtech stack at Google, that claim is a mischaracterization of Google's systems."[59] He further contends that "Google's code base is not generally made up of microservices."[60] I note here that were this the case, migrating a single monolithic binary with internal dependencies could well be more straightforward than migrating a distributed system, because the dependency graph for a monolithic system is local, well-defined, and can be managed in a single codebase with coordinated build and release tooling.

38. Regardless, the processes I describe in my Opening Report (e.g., stubbing, traffic mirroring, shadow deployment, and controlled rollout) to analyze and test specific functionalities while migrating apply to monolithic inter-dependent architectures. More tightly-coupled, or even monolithic, structures can also be migrated using the Strangler Fig pattern.[61] The framework I put forward[62] is designed to flexibly support both paradigms, as it discusses how a monolithic workload can be handled as well through a re-architect migration as needed.

39. Mr. Maymudes claims that I "ignore[] both the non-serving aspects of GAM (e.g., its UI and offline processing components) and the many other [bespoke systems that are not specific to GAM]... [such as] Google's logging system, Billing3, and so on,"[63] and Dr. Nieh claims that I "rel[y] on a

[59] Maymudes Rebuttal Report, ¶ 33.
[60] Maymudes Rebuttal Report, ¶ 56.
[61] *See* Chris Richardson, "Refactoring a monolith to microservices," Microservice Architecture, accessed August 7, 2025, https://microservices.io/refactoring/ ("Truly greenfield development of microservices-based applications is relatively rare. Many organizations that want to adopt microservices already have a monolithic application. As I describe in chapter 13 of my book, the recommended approach is to use the Strangler application pattern and incrementally migrate function from the monolith into services.").
[62] I use Google's migration guide to show that there are standard steps for migrations that largely align with my proposed schedule. *See* ¶ 25.
[63] Maymudes Rebuttal Report, ¶ 34 ("The Bjedov Report's analysis of a divestiture of DFP or AdX tends to focus on the ad serving aspects of GAM, but often ignores both the non-serving aspects of GAM (e.g. its UI and offline processing components) and the many other Google systems on which GAM depends. Based on my experience with DV360, Google's ad tech products are dependent on a host of broader proprietary Google systems that support other Google products beyond those that are specific to its ad tech systems. Indeed, Professor Nieh illustrates in his opening report that many bespoke systems that are not specific to GAM (e.g., developed and maintained by different

surface level analysis of dependencies."[64] I do not ignore dependencies that are non-specific to AdX and DFP, and in fact I specify that "the team of engineers involved in the analysis stage of the migration will need to follow standard processes to identify the components relevant to the migration and understand *all relevant dependencies*."[65] Mr. Maymudes simply provides examples of some of the dependencies that will be mapped in Stage 1 of the migration process.

40. Mr. Maymudes further suggests that I equate a system of ten microservices with that of thousands.[66] This is not the case. Rather, I specifically emphasize the importance of coordinated migration and testing when evaluating a large system with many dependencies.[67]

41. Mr. Maymudes also misconstrues how work on microservices is carried out in parallel. He states that I "assume [work on any microservice] can be done in parallel [to any other]."[68] This is not the case. In fact, I state that "[t]esting [a software system] composed of thousands [of microsystems], as in the case of AdX or DFP, requires coordinated validation…staged rollout tooling…[and] pre-production load tests."[69] I agree that testing services requires integration testing, which is why I suggest both a staged approach to swapping individual services (using the Strangler Fig pattern of

---

teams unrelated to the GAM engineering team with a different employee hierarchy, different resources, etc.) provide crucial services that GAM could not operate without, including Google's logging systems, Billing3, and so on.").

[64] Nieh Rebuttal Report, ¶ 57 ("Because she relies on that surface level analysis of dependencies, Dr. Bjedov makes generalized claims about what would 'accelerate migration processes,' such as 'high test coverage [that] also enables rapid iteration and verification during the testing and verification stage of migration.'").

[65] Bjedov Opening Report, ¶ 172 ("Finally, the team of engineers involved in the analysis stage of the migration will need to follow standard processes to identify the components relevant to the migration and understand all relevant dependencies.").

[66] Maymudes Rebuttal Report, ¶ 58 ("Even though Dr. Bjedov appears to recognize that a larger system would demand more complexity and time to migrate, her estimated divestiture timelines do not take that into account, as she bases her estimates for AdX or DFP divestiture on how long migrations 'typically' take without any meaningful reference to relative size or complexity other than the cursory word 'large-scale.' Based on my professional awareness of many large software projects at Google and other companies, including the two significant software migrations relating to Google's ad tech stack for which I have personal, first-hand experience leading (discussed elsewhere in my report), and which were of smaller scale and complexity than divesting AdX or DFP, Dr. Bjedov's time estimates should not be based on how long migrations 'typically' take in her view and are far too short as a result.").

[67] Bjedov Opening Report, ¶ 69 ("Based on my experience, testing an application composed of 10 microservices is straightforward. Testing one composed of thousands, as in the case of AdX or DFP, requires coordinated validation, but it falls well within standard practices at Google and any other large technology company. These systems are already instrumented with testing frameworks and staged rollout tooling.").

[68] Maymudes Rebuttal Report, ¶ 53 ("Dr. Bjedov does not account for the possibility that the process of rewriting code to use replacement services or recreating services may have interdependencies among different services that make it impossible for all the projects to be done simultaneously. She appears to instead assume that 'work on each service' — namely, the numerous services that AdX and DFP rely on—'can be undertaken [in] parallel.'").

[69] Bjedov Opening Report, ¶ 69.

slowly passing traffic through the new service)[70] as well as holistic integration testing after all services have been replaced.[71] I discuss in Section VI.B.2.c and VI.B.2.d why integration testing represents an additional phase of testing, as each service is also tested independently and gradually in an integrated environment. The iterative, gradual process I lay out for each service shows that migration of microservices can occur in parallel and in coordination with other services.

42. Lastly, Mr. Maymudes is incorrect that I assume that all projects would be completed simultaneously.[72] Analyses can be done in parallel, and the workstreams in stage two would coordinate with each other, but they would not necessarily start and stop their work at the exact same time. My plan and timeline account for some degree of inter-dependencies between systems, and I include coordination and discussion between parallel workstreams and migration efforts for each service to address this.[73] The approach and timeline I propose in my Opening Report incorporate the realities of migrating such a complex architecture through mapping dependencies, creating mocks and stubs (which facilitate work to be done in parallel even despite interdependencies), and staging gradual deployments of replacement services[74] that can and should occur in parallel to one another.

2) I Account for Planning, Service Evaluation, and Rewriting in the Proposed Timelines

[70] Bjedov Opening Report, Section IX.C.
[71] Bjedov Opening Report, Section IX.D.
[72] Mr. Maymudes characterizes AdX and DFP as "highly complex, entwined, and integrated," which does not align with Dr. Weissman's assessment of the code. Maymudes Rebuttal Report, ¶ 53 ("Moreover, Dr. Bjedov does not account for the possibility that the process of rewriting code to use replacement services or recreating services may have interdependencies among different services that make it impossible for all the projects to be done simultaneously. She appears to instead assume that 'work on each service'—namely, the numerous services that AdX and DFP rely on—'can be undertaken [in] parallel.'").
[73] Bjedov Opening Report, ¶ 189 ("Each service can be analyzed in parallel by its own two-to-three-person engineering pod. Pods may own the code inspection, build a stub (or full replacement), define tests (see Section IX.D), and shepherd the change through canary and production ramps. Coordination can happen in a regular "integration strike-team" forum so architectural decisions remain coherent across the migration."); *see also* Bjedov Opening Report, ¶ 166 (Tables IX.1–2 summarizing the parallel workstreams I propose).
[74] Bjedov Opening Report, Section IX.C.

43. Mr. Maymudes and Dr. Nieh state that I skip several essential steps in my migration timeline, especially around planning, service evaluation, and rewriting.[75] They are incorrect. I account for these steps in my proposed workplan.

a) My Proposal Accounts for Planning

44. Dr. Nieh's implication that planning a migration of AdX or DFP will necessarily span many months or remain unbounded is inconsistent with how modern engineering teams execute well-scoped migrations (see Section III.A).[76]

45. Mr. Maymudes states that I omit elements of the planning process from my migration timeline to determine decisions around business and regulatory requirements.[77] I discuss the assumptions I make for steps that precede my analysis in Section III.C. In general, I see no reason why the acquirers of AdX and DFP would not be expected to maintain continuity with existing business logic and compliance obligations[78] that are in line with industry standards, unless the acquirer deliberately chooses to modify them. Other AdTech companies exist and adhere to industry standards and regulations.[79]

---

[75] Maymudes Rebuttal Report, ¶ 46 ("Dr. Bjedov omits multiple important planning steps that must precede a deployment analysis…Similarly, because I understand that Plaintiffs request AdX and DFP be divested to separate buyers, a planning stage would require creating a plan for separation of currently intertwined AdX and DFP code—the execution of which Dr. Bjedov also omits from her migration plans. And even within deployment analysis, Dr. Bjedov omits actual analysis of the dependencies AdX and DFP rely on beyond just identifying them. In other words, she omits the planning and analysis that would need to be done to determine how to replicate those dependencies in the target infrastructure."); *see also* Nieh Rebuttal Report, ¶ 139 ("Dr. Bjedov provides an outline of her proposed stages for a divestiture of AdX and DFP in Tables IX.1 and IX.2 of the Bjedov Report. Both of those tables are missing key components that would be required to actually implement her proposed workflow, and the components that she does account for are in an ordering that would not result in a working system at any intermediate stage of the migration.").

[76] Nieh Opening Report, ¶ 159 ("Based on the requirements of the AdX and DFP functionalities (just a subset of which were outlined above) and the complex, sophisticated hardware and software infrastructure on which they operate, I expect producing an architecture development plan would take a considerable amount of time.").

[77] Maymudes Rebuttal Report, ¶ 46 ("For example, a divestiture of AdX or DFP would demand significant decisions about the business and regulatory requirements of the divested products, particularly because AdX and DFP currently do not exist as either standalone codebases or product offerings to customers.").

[78] I note again that functional equivalence in system outcome does not require exact re-implementation of Google's systems.

[79] *See, e.g.*, "Advertising, Data Protection and Privacy," Amazon Ads, accessed August 7, 2025, https://advertising.amazon.com/resources/ad-policy/eu-data-protection-and-privacy ("[Amazon systems and processes] meet our own high standards and the requirements of all applicable data protection laws, including the General Data Protection Regulation (GDPR) and the Privacy and Electronic Communications Directive (ePrivacy Directive). Amazon Ads services comply with the requirements of the GDPR, the ePrivacy Directive and other applicable law."); "Privacy Policy," Criteo, last modified April 4, 2025, https://www.criteo.com/privacy/ ("Criteo

46. With the lift-and-shift approach to migration, the expectation is that significant changes should not be needed for business rules or practices.[80] In the few cases where changes or review might be needed, the acquirer can work with Google to evaluate any decisions that need to be made and adjust the timeline accordingly.[81]

b) Evaluating Appropriate Replacements is Part of Technical Decoupling

47. Both Mr. Maymudes and Dr. Nieh raise concerns about the complexity and necessity of rewriting services. For example, Mr. Maymudes suggests that an acquirer would have to "build new systems" to replace internal dependencies,[82] but this ignores the fact that an acquirer will evaluate alternative systems (such as publicly available alternatives or its own proprietary systems) and determine its preferred approach to replacing the dependencies.

48. Mr. Maymudes suggests that I have forgotten about dependencies and that this will cause the problem during migration. On the contrary, dependencies are resolved in Stage 2 of the process. The goal of decoupling can remove interactions both between AdX and DFP, and between AdX (and later DFP) and underlying Google internal dependencies. In their place should be new connections to the systems of the acquirers' choice that provide functional equivalence with

---

participates in the IAB Europe Transparency & Consent Framework and complies with its Specifications and Policies."); "Privacy and The Trade Desk Platform," The Trade Desk, last modified February 28, 2025, https://www.thetradedesk.com/legal/privacy ("The Trade Desk is a member in good standing of the Network Advertising Initiative (NAI). The Trade Desk also follows the industry self-regulatory guidelines of the Digital Advertising Alliance, the Digital Advertising Alliance of Canada, and the European Digital Advertising Alliance. The Trade Desk also implements and adheres to the specifications and policies of the IAB EU Transparency & Consent Framework as part of our compliance with EU data protection law.").

[80] "Rehosting Mainframe Applications on the Sun Platform," Sun Microsystems, (2001): 1–14, at 6 https://www.cambex.com/products/mfrehosting/Whitepapers/Sun%20rehosting.pdf ("The embedded business rules of the customer's mainframe applications, which define the business processes and provide competitive differentiation, are completely preserved, so no changes to business processes or practices are required. With Sun rehosting solutions, end users typically experience no change to their interaction with familiar applications.").

[81] *See* Bjedov Opening Report, ¶ 68 ("Based on an analysis of traffic patterns, replication needs, and regulatory constraints, a monitor or technical committee, along with Google and the acquirer, can determine the number and placement of servers needed to support AdX or DFP on GCP. Google would also be able to support a monitor or technical committee using its current expertise in operating AdX and DFP.").

[82] Maymudes Rebuttal Report, ¶ 53 ("Dr. Bjedov's omission significantly undermines her timeline estimate because creating a version of AdX or DFP that can function outside Google's infrastructure would require building new systems, a task of considerable complexity. For example, recreating even a single system like F1, Kansas, or Napa, that has no replacement outside Google could take years."); *see also* Nieh Rebuttal Report, ¶ 8 ("The existence of the proprietary dependencies that AdX and DFP rely on, and the complexity associated with rewriting or removing them, belies Prof. Weissman's claim that migrating AdX and/or DFP would require no major software rewrite or redesign.").

acceptable performance. These dependencies are what Prof. Weissman refers to as first-order dependencies.[83] Once those are successfully replaced, higher order dependencies are not relevant to the migration process.

49. As I state in my Opening Report, the acquirer will evaluate all services and choose the best option, which may be keeping the dependency as is, replacing with a functionally equivalent service, or rewriting as needed.[84] If indeed "recreating [a dependency] could take years,"[85] then the acquirer can choose to replace the system with an appropriate alternative, as I state in my Opening Report.[86] I discuss the integration of these replacement dependencies into AdX and DFP application code in Section III.B.2.c.

50. I consider that AdX and DFP currently rely on internal Google services[87] in the planning and evaluation stages of my timeline, where the purpose of those stages is to create a thorough catalog and dependency map of each service that can be used to determine what the appropriate alternative for dependencies is when needed.[88] As I state in my Opening Report, I agree with Dr. Nieh and Mr. Maymudes that this is a substantial undertaking.[89] This is why I suggest each service be evaluated by a dedicated engineering pod with coordination across teams so that integration and architectural decisions remain coherent.[90]

---

[83] Weissman Rebuttal Report, ¶ 37 ("When discussing dependencies, I use the term first-order to refer to the immediate dependency of a system (e.g., immediate dependencies of AdX and DFP), and terms such as second-order and third-order to refer to indirect dependencies (i.e., dependencies further down the chain).").
[84] Bjedov Opening Report, ¶ 191 ("Building on the findings from stage one, this workstream evaluates all systems, services, and tools used by the product for its internal analysis and communication and determines if each interaction can stay as-is, must be rewritten, or demands a full replacement.").
[85] Maymudes Rebuttal Report, ¶ 53 ("For example, recreating even a single system like F1, Kansas, or Napa, that has no replacement outside Google could take years.").
[86] Bjedov Opening Report, ¶ 192 ("Where a service to be replaced is proprietary, a small team can conduct a planning exercise to select a substitute and documents its pros, cons, and migration steps.").
[87] Nieh Rebuttal Report, ¶ 77 ("Dr. Bjedov provides no alternatives to and is unable to draw any so-called 'similarities' between GCP services to F1, Napa, SmartASS, various security and privacy-related systems such as UIS and Stargate, Kansas, and so on—most of the bespoke Google systems and infrastructure that I list in Appendix 2 of my Opening Report.").
[88] Bjedov Opening Report, ¶ 172 ("Finally, the team of engineers involved in the analysis stage of the migration will need to follow standard processes to identify the components relevant to the migration and understand all relevant dependencies.").
[89] Bjedov Opening Report, ¶ 196 ("One of the most technically complex aspects of this system migration involves reconciling differences in configurations and capabilities between the source and target systems.").
[90] Bjedov Opening Report, ¶ 189 ("Each service can be analyzed in parallel by its own two-to-three-person engineering pod. Pods may own the code inspection, build a stub (or full replacement), define tests (see Section IX.D), and shepherd the change through canary and production ramps. Coordination can happen in a regular 'integration strike-team' forum so architectural decisions remain coherent across the migration.").

(i) Similarity Between Google's Internal Environment and GCP Supports a Viable Migration Path

51. Before an acquirer can begin migrating AdX or DFP, it will need to evaluate and select a suitable target environment. In my Opening Report, I suggest that GCP would be a logical migration target to minimize differences between Google's internal environment and one that can be fully controlled by an acquirer.[91] Dr. Nieh and Mr. Maymudes both discuss ways in which Google's internal environment differs from Google Cloud Platform,[92] and Dr. Nieh goes on to describe why migration between Google's internal environment to GCP would be challenging.[93] I do not dispute that there are differences between these environments and that it will take work to migrate from one to the other. However, Dr. Nieh and Mr. Maymudes overstate the differences between Google's internal environment and GCP, and consequently misrepresent the difficulty of migrating AdX or DFP to GCP.

52. In his Rebuttal Report, Dr. Nieh argues that my observation that a migration of DFP and AdX would be made easier by the original and destination environments being similar amounts to my "implicitly agreeing with [Dr. Nieh's] opening report that moving DFP or AdX from one environment to another environment is a challenging problem that would be somewhat alleviated if the original and destination environments were as similar as possible."[94] Dr. Nieh and I are indeed in agreement: It is my opinion that a migration is easier the more similar the target environment is to the original environment.

53. As discussed in Section III.B, the tools available in the target environment do not need to be exact replacements. Dr. Nieh argues that the absence of direct access to Google's internal tools in GCP

[91] Bjedov Opening Report, ¶ 30 ("An acquirer of AdX or DFP could migrate these applications first to Google's platform, GCP, as the closest available public cloud to Google's private cloud, to minimize differences in infrastructure with Google's private cloud.").
[92] Maymudes Rebuttal Report, ¶ 115 ("The difficulty of migrating DFP or AdX is also compounded and made more complex by the nature of a divestiture, which would require migrating the full feature set of DFP or AdX, and all of its customers, off of Google's internal infrastructure on to a different software and hardware infrastructure and environment. That challenge also applies if a divestiture buyer plans to migrate DFP or AdX to Google Cloud Platform ('GCP') because, contrary to Dr. Bjedov claims of GCP's similarity to Google's internal infrastructure, GCP's cloud computing environment and infrastructure is very different from Google's internal infrastructure."); *see also* Nieh Rebuttal Report, Section IV.
[93] Nieh Rebuttal Report, Section IV.
[94] Nieh Rebuttal Report, ¶ 70 ("I also note that by claiming such, Dr. Bjedov appears to be implicitly agreeing with my opening report that moving DFP or AdX from one environment to another environment is a challenging problem that would be somewhat alleviated if the original and destination environments were as similar as possible.").

does not simplify the migration of AdX or DFP.[95] This argument misconstrues how Google Cloud is designed. The relevant question is not whether the acquirer can access the same internal version of tools as Google, but whether GCP exposes managed services that offer the same functionality and provide the same runtime guarantees, which it does.[96]

54. GCP does not have "drop-in replacements"[97] for Google's internal systems, but similarities between GCP and Google's internal systems facilitate easier migration compared to other public cloud platforms or private cloud services. I discuss how GCP offers a strong target environment through robust data and deployment infrastructure in Section III.B.2.f.

55. In addition, Dr. Nieh states that I "ha[ve] not shown that GCP data centers exist in the same physical locations as all of the data centers for Google's internal infrastructure, including those that [AdX and DFP] specifically rel[y] on for serving and offline processing."[98] However, GCP's global infrastructure footprint overlaps with Google's internal production Points of Presence or "POPs," as Google Cloud operates in 42 regions, with 127 zones, and 202 edge locations.[99] This

[95] Nieh Rebuttal Report, ¶ 74 ("GCP abstracts away Google's internal infrastructure so that GCP customers do not have direct access to Google's internal infrastructure, so a divestiture buyer could not simply just copy GAM software that depends on, for example Borg, to GCP. Put another way, Borg is not available to GCP's customers, so GAM's dependency on Borg would have to be rewritten or somehow otherwise addressed."). *See generally* Nieh Rebuttal Report, ¶¶ 83–87.

[96] Kubernetes is functionally an open-source version of Borg that was originally developed at Google as an open-source abstraction of Borg, and which now serves as the foundation for GKE (Google Kubernetes Engine), a production-grade orchestration layer that preserves Borg's key scheduling, repair, and resource-packing logic. These concepts have been hardened in production over two decades and are now exposed in a secure, scalable, and accessible form through GCP. Borg streamlines deployment for Google-specific software, but this does not mean that open-source alternatives (like Kubernetes) cannot serve the same function. Craig McLuckie, "From Google to the world: The Kubernetes origin story," Google Cloud, July 22, 2016, https://cloud.google.com/blog/products/containers-kubernetes/from-google-to-the-world-the-kubernetes-origin-story*; see also* Dean Hildebrand and Denis Serenyi, "Colossus under the hood: a peek into Google's scalable storage system," Google Cloud, April 19, 2021, https://cloud.google.com/blog/products/storage-data-transfer/a-peek-behind-colossus-googles-file-system ("[Borg, Spanner, and Colossus are] used to provide the underlying infrastructure for all Google Cloud storage services, from Firestore to Cloud SQL to Filestore, and Cloud Storage."); *see also* Declaration of Samuel Greenfield, July 28, 2025 (Cloud BigTable provides "comparable functionality" to Bigtable).

[97] Nieh Opening Report, Appendix 2.

[98] Nieh Rebuttal Report, ¶ 146.

[99] All of which are connected by 3.2 million kilometers of Google-owned fiber. *See* GOOG-AT-MDL-004210393, at -432 ("We've been investing in our own private network connections across the globe for more than a decade. So when you use Cloud Storage to globally replicate data, that data isn't transmitted over the public internet."); *see also* "Cloud locations," Google Cloud, accessed August 7, 2025, https://cloud.google.com/about/locations/ ("This continually-growing network connects our infrastructure globally to more than 3.2 million kilometers of terrestrial and subsea fiber - 10 times the reach of the next leading cloud provider."); "What's in a name? Understanding the Google Cloud network 'edge'," Google Cloud, February 22, 2021,

4) Shut down the old systems once traffic is fully moved over to the new system.

95. As I state in my Opening Report, internal software migrations are routine in industry,[220] including at companies like Facebook and Google.[221,222] In fact, Google's lead of the GAM engineering team, Noam Wolf, himself stated that he's done "a lot of migrations."[223] The fact that internal software migrations are routine for Google, suggests that Google has the background and tools to perform functions necessary for the software migration of AdX or DFP.[224]

2) Software Migrations and Software Deployments Have a Very Close Relationship

96. Dr. Nieh also states that I conflate software deployments with software migrations, and that software deployments are irrelevant for an AdX or DFP migration.[225] Dr. Nieh misunderstands the examples of deployments that I provide and how they required software migrations. Specifically, while deployment is always a component of migration,[226] migration work can also be a component of deployment.[227]

---

[220] Bjedov Opening Report, Section IV.B.

[221] I was part of countless internal software migrations during my tenure at Facebook. *See* Bjedov Opening Report, Section V.B.5, where I describe the Look Back videos migration; *see also* Bjedov Rebuttal Report, Section IV.D.1, where I discuss the AdTech stack rewrite and migration.

[222] Maymudes Rebuttal Report, ¶ 8 ("From 2013 to 2018, I was Technical Lead for another large-scale migration known internally as 'Skyray,' which involved migrating Xbid onto infrastructure shared with Google Ads, internally referred to as the CAT2 serving stack. The Skyray migration was entirely within Google's production environment, such that Xbid could rely on Google's infrastructure systems both before and after the migration.").

[223] Deposition of Noam Wolf (Google), July 18, 2025, 235:12–236:8 ("Q. We've moved on from open source now. [redacted]").

[224] *See generally* Bjedov Opening Report, Section IV.B, where I discuss routine migrations in industry and the benefits of the practice.

[225] Nieh Rebuttal Report, ¶ 91 ("Dr. Bjedov refers to certain projects or work as large-scale software migrations in order to suggest they are comparable to the type of migration required for an AdX or DFP divestiture, but what she is actually referring to are software deployments. That distinction is critical because *deployments* do not involve the same considerations and technical complexity that a migration of AdX or DFP would.") (emphasis in original).

[226] *See* Bjedov Opening Report, Section IX.E, where I describe the "Final Deployment" stage; *see also* "Plan your migration," Microsoft Learn, last modified August 1, 2025, https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/migrate/plan-migration ("A migration plan defines the specific order, timing, and approach for migrating workloads to Azure. This plan translates high-level migration strategies into actionable deployment sequences.").

[227] For example, Look Back videos, which required a lot of deployment work, also required migration work. I provide more details on this work below.

97. Even deployments of new products or features can involve migrations. For example, Dr. Nieh critiques my example of Facebook's "Look Back" as being a deployment, not a migration,[228] incorrectly assuming that Look Back did not involve moving software across environments.[229] However, he ignores the fact that in this project, many software migrations occurred behind the scenes to allow the product to run, including migrating photos between servers and to different storage formats, migrating the code to different compute pods, ensuring the landing zone storage subsystems and permissions were correct, and testing and debugging the pipeline.[230] Thus, even in the case of Look Back videos, which Dr. Nieh describes as a deployment of a new product, migration work was required that is relevant to the migration of AdX or DFP. Dr. Nieh conflates product improvements with core migration tasks.

98. While Dr. Nieh presents a restrictive definition of migration, as I discuss in Section IV.A.1, he simultaneously argues that extraneous work done outside of migration work should also be included in the migration timeline. In Section V.B of his Rebuttal Report, Dr. Nieh critiques the timelines of the industry migration examples that I presented in my Opening Report. In arguing that the migration timelines I presented are not viable, Dr. Nieh frequently interprets non-migration related work such as prior product improvements, future data center scaling, financial analysis, and exploratory development work as part of a software migration's timeline. I discussed this further in Section V.A of this report, but I note that Dr. Nieh's broad interpretations in Section V.B of his Rebuttal Report further demonstrate his non-standard definition of software migrations.

---

[228] Nieh Rebuttal Report, ¶ 92 ("Deploying software refers to releasing software, which in the context of Google's ad tech would generally mean releasing a new feature that runs on Google's internal infrastructure, i.e., there is no migration to a completely new or different infrastructure that lacks the full set of proprietary dependencies available as part of Google's internal infrastructure. In contrast, a large-scale software *migration* in the context of a divestiture refers to moving software to a new environment, and all the work required to make the migrated software operational in that new environment.") (emphasis in original).

[229] Nieh Rebuttal Report, ¶ 95 ("Dr. Bjedov lists eight examples that she refers to as demonstrating 'successful migration of complex large-scale software applications across different environments,' excluding her experience with deploying the Look Back feature on Facebook, which did not involve moving across environments.").

[230] For example, to ensure there was sufficient compute power available, we had to bring back online and reprovision a whole cluster of servers. These servers were up for a week and then went through the decommissioning process again. To get the type of access to the user photos that was needed, user photos had to be migrated out of the standard format in haystack and temporarily moved to differently provisioned storage that could provide the throughput the compute servers needed to create videos. Additional high-throughput storage was provisioned as a landing place for the videos and the pipeline was tested and debugged. For more on delaying decommissioned servers, *see* @Scale, "Look Back Videos," YouTube Video, November 12, 2014, 12:00, https://www.youtube.com/watch?v=BFMuilYacLQ.

B. Mr. Maymudes Faults My Report for Not Covering Issues Within Dr. Weissman's Scope

99. Mr. Maymudes claims that I do not apply a complexity analysis of the AdX and DFP migrations from a software engineering perspective.[231] This mischaracterizes both the scope of my report, and my professional experience as a capacity engineer who has participated in many complex software migrations over the course of my career. Furthermore, Mr. Maymudes ignores the fact that Dr. Jon Weissman conducted a complexity analysis, which serves as the foundation for my migration assessment in my Opening Report.[232]

100. As a result, Mr. Maymudes concludes that I ignore the ostensible fact that, "the existing source code, even if recompiled, simply will not function outside of Google's proprietary production computing environment and would thus need to be substantially rewritten before it could be 'migrated' to a 'typical cloud environment.'"[233] Dr. Weissman opines that while some rewrites are necessary, Dr. Nieh overstates the extent of the rewrites. I rely on Dr. Weissman's analysis and considered the necessary rewrites when developing my migration timeline,[234] as discussed in Section III.B.2.

101. Moreover, my analysis reflects a systems-level operational perspective informed by my extensive work with Site Reliability Engineers (SRE) and Performance Engineers over a decade. As a result, I specifically address how risk-mitigation techniques and carefully managed migration processes help manage issues that can arise during software migrations. This is just as critical to migration success as any changes to the source code. Having personally overseen complex

[231] *See* Maymudes Rebuttal Report, ¶ 29 ("Engaging with the complexity of a divestiture of DFP or AdX requires an analysis that is grounded in understanding the complexity and scale of Google's ad tech systems from at least both the perspectives of Software Engineering and Site Reliability Engineering. However, Dr. Bjedov appears to narrowly frame her opinions based on her experience in SRE-adjacent roles. She primarily focuses on certain aspects of scale in her evaluation of a potential divestiture of DFP or AdX, and does not holistically address the complexity of developing and maintaining large-scale, distributed software systems throughout their software lifecycles.").

[232] *See* Weissman Opening Report, ¶ 2 ("I have been asked to assess whether it is feasible, from a technical perspective, to copy and migrate certain technical assets, such as repositories of source code, configuration files, and data, from Google's Ad Exchange ('AdX') and current publisher-ad server ('DFP'), including the final auction logic and remainder of DFP. I have been asked to assess the technical feasibility of developing or adapting application programming interfaces ('APIs') that facilitate the proposed data sharing and interim remedies.").

[233] Maymudes Rebuttal Report, ¶ 32.

[234] *See* Bjedov Opening Report, ¶ 191 ("Building on the findings from stage one, this workstream evaluates all systems, services, and tools used by the product for its internal analysis and communication and determines if each interaction can stay as-is, must be rewritten, or demands a full replacement.") and ¶ 196 ("Some adjustment of configuration parameters or possibly rewriting parts of the application code could be required in this work.").

software migrations in production environments, I can assess Dr. Weissman's report, as well as internal documents at Google that discuss the migration of AdX and DFP, and translate abstract engineering tasks into a migration timeline. I provide additional discussion about my methodology in Section III.A.2.

C. Dr. Nieh Mischaracterizes the Applicability of Software Migration Tools

102. Dr. Nieh states that some of the tools I list in my Opening Report are "irrelevant to software migrations of the kind required [for AdX and DFP]."[235] Specifically, he states that Anthos (a Google Cloud product),[236] containers,[237] Network Analyzer, Database Migration Assessment tool, Flow Analyzer, and Visor all do not have "any relation to a divestiture of DFP or AdX."[238]

103. Dr. Nieh misunderstands the purpose of my examples, which is to show that there are tools available to assess the migration complexity of software application deployments,[239] not necessarily to give the exact tools that an acquirer would use, as the acquirer may select from a range of tools, which I describe the relevance of for several examples below.

104. For example, Network Analyzer is responsible for mapping network topology, which is vital for engineers to ascertain what must be rebuilt or refactored in the target environment prior to migration.[240]

105. Connectivity test suites, which act like synthetic end users, are run before, during, and after a migration to ensure things such as firewall rules and DNS entries are met. They are used to

---

[235] Nieh Rebuttal Report, ¶ 132.
[236] "Anthos," Google Cloud, accessed August 7, 2025, https://cloud.google.com/anthos?hl=en ("Get access to Anthos capabilities through Google Kubernetes Engine (GKE) and Google Distributed Cloud (GDC).").
[237] Nieh Rebuttal Report, ¶ 134 ("Containers and multi-cloud container platforms like Anthos assist SWEs with packing up known dependencies into containers that can then be deployed.").
[238] Nieh Rebuttal Report, ¶ 135.
[239] *See* Bjedov Opening Report, Section IV.B.4 ("Many Tools are Available to Assess Migration Complexity of Software").
[240] *See* "Network Analyzer overview," Google Cloud, last modified August 7, 2025, https://cloud.google.com/network-intelligence-center/docs/network-analyzer/overview ("Network Analyzer…provides insights about network topology"); "What is Skydive," Skydive, accessed August 6, 2025, https://skydive.network/ ("Skydive is an open source real-time network topology and protocols analyzer providing a comprehensive way of understanding what is happening in your network infrastructure.").

confirm the status for each application instance and prove that applications are maintaining access to the connections they need as the migration progresses.[241]

106. DMA tools scan databases to monitor for any data-related issues that may arise during the migration.[242] Flow analyzer tools capture network traffic to reconstruct communications between tools, providing engineers with a dependency map which guides their development of things such as security rules and routing tables in the new environment.[243] As I discuss in Section IV.A, because of its frequent internal software migrations, Google likely has the necessary background and tools to carry out the software migration of AdX and DFP.

V. DR. NIEH AND MR. MAYMUDES MISCHARACTERIZE INDUSTRY EXAMPLES IN WAYS THAT INFLATE THE TIMELINE OR DIFFICULTY OF MIGRATION

107. Dr. Nieh and Mr. Maymudes dispute the relevance of, and often mischaracterize, many of the industry examples I present in my Opening Report.[244] Their mischaracterizations are at odds with my long career facilitating software migrations. This includes my time as a capacity engineer at Facebook where I worked on many migrations,[245] including the migration of Instagram and

[241] *See* Daniele Barilla and James Bland, "Achieve testing success with the AWS Application Migration Service for painless and simplified cut-overs," AWS Cloud Operations Blog, July 28, 2023, https://aws.amazon.com/blogs/mt/achieve-testing-success-with-the-aws-application-migration-service-for-painless-and-simplified-cut-overs/ ("integrations with other services and backup servers could cause unexpected behaviors during your testing and cut-over…The Application Migration Service uses continuous block-level replication technology making it easy to Re-host / Lift and Shift servers to AWS…Application Migration Service prevents a final cutover until you've gone through a testing cycle.").

[242] *See* "Readiness Check Utility Overview," Google Cloud GitHub, accessed August 10, 2025, https://googlecloudplatform.github.io/database-assessment/user_guide/readiness_check/overview/ ("Running the readiness check before initiating a migration can help you proactively address configuration problems, ensuring a smoother migration process. It checks various aspects of your source database against the requirements and recommendations for the target Google Cloud database service.") (emphasis omitted); Celia Antonio and Pritesh Jani, "Save time, money and modernize your legacy database estate …. but first assess," Google Cloud, July 12, 2023, https://cloud.google.com/blog/products/databases/introducing-open-source-database-migration-assessment-tool/.

[243] Eric Graham, "Using Flow Data to Optimize Firewall Rules: Best Practices and Cloud Migration Insights," ElastiFlow, July 22, 2025, https://www.elastiflow.com/blog/posts/using-flow-data-to-optimize-firewall-rules-best-practices-and-cloud ("Flow data, such as NetFlow, IPFIX, and sFlow, provides a granular view of network communications…[and] becomes a powerful asset for: Auditing existing firewall rules; Identifying redundant or obsolete allow lists; Tracking application dependencies; Understanding if applications are successfully communicating bidirectionally; Safely tightening security postures during cloud migrations").

[244] Nieh Rebuttal Report, Section V; Maymudes Rebuttal Report, Section IV.

[245] On average, I was typically involved in 2 large-scale migrations per year as Facebook built out new regions. This is along with countless smaller migrations building and moving development servers; from 2014 onward, I "owned" the service clusters and was in charge of migrating the dozens of individual services for each of these clusters.